NATIONAL IMAGERY TRANSMISSION FORMAT STANDARD (NITFS)
REQUEST FOR CHANGE (RFC)

RFC CONTROL NUMBER 95-015A
(To be filled in by NTB Secretary)

DATE SUBMITTED 11/22/94 DATE RECEIVED 11/28/94

ORIGINATOR Dan Challstrom
TELEPHONE (215) 531-6164

MAILING Martin Marietta
ADDRESS

ORGANIZATION TYPE. user

PRIORITY routine

FUNCTION operational

DOCUMENT NUMBER MIL-STD-188-198A PAGE
DOCUMENT NITFS JPEG PARAGRAPH 5.2.3.3.5.5.4

PROBLEM DESCRIPTION
This change provides a mechanism to embed FEC (Forward Error Correction) codes inside of the NITF JPEG Format.

RECOMMENDED WORDING
See attached.

RATIONALE
This change provides protection to the crucial portions of the frame and scan levels within the JPEG format.

REMARKS
Approval pending validation by the JITC.

TOTAL COST OF IMPLEMENTATION

PROPOSED TIMEFRAME OF IMPLEMENTATION
next revision

ANTICIPATED USER IMPACT

NTB REVIEW DATE 12/8/94 NTB RECOMMENDATION approved as modified
SUBSTANTIVE ISSUES

DATE SUBMITTED TO ISMC
ISMC REVIEW DATE

DATE SUBMITTED TO DISA

ISMC DECISION

IMPLEMENTATION DATE

NITFS-CCB FORM 1(REVISION 3)

5.2.3.3.5.5.4 NITF APP_9 Forward Error Correction (FEC) Code

The NITF APP_9 application data segment contains FEC (Forward Error Correction) codes which are used to protect NITF-JPEG/DCT header and table data from bit errors.

The FEC codes are applied to:

1. NITF/JPEG/DCT Frame Header and Misc. Tables

2. NITF/JPEG-DCT Scan Header and Misc. Tables

Two different forms of the APP_9 application data segment shall be used for each image block, one for the frame data and one for scan data. The placement of these two forms of APP_9 application segments is shown in Figures XX and XXI.

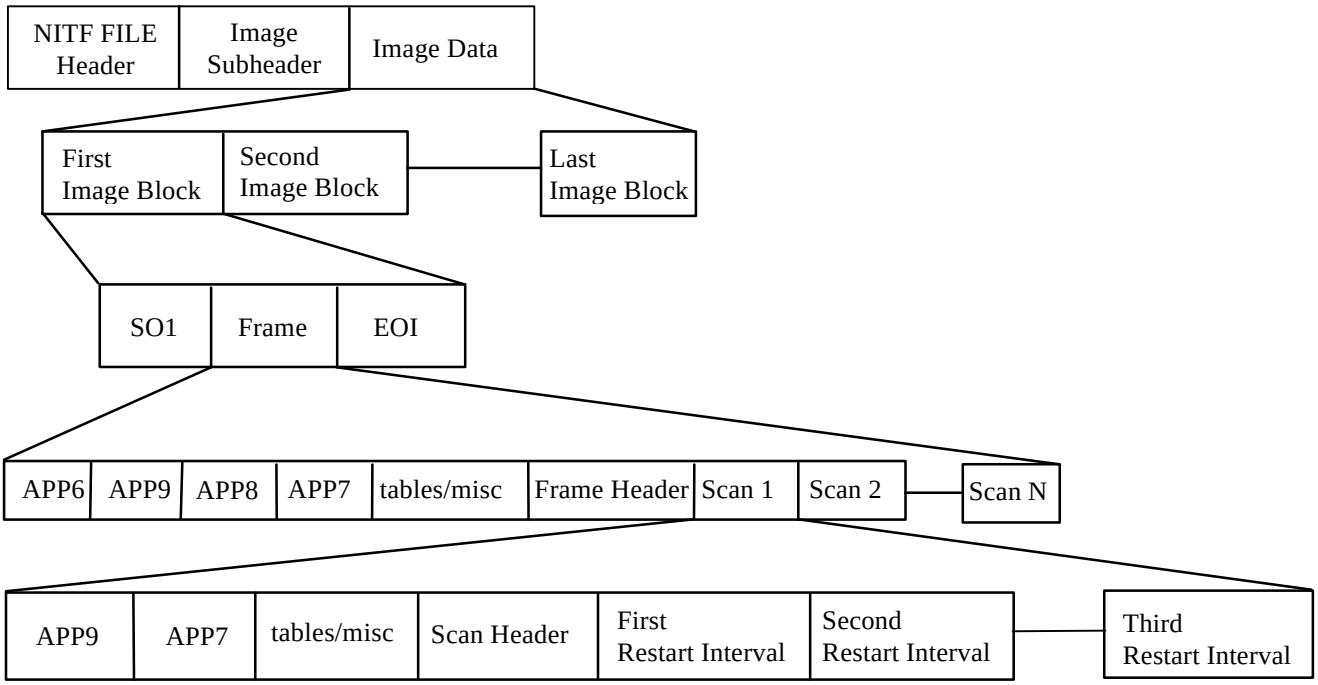

FIGURE XX. NITF 12 BIT JPEG/DCT MULTIPLE BLOCK FILE STRUCTURE WITH FEC (IMODE=B OR P)

Table XXII NITF-JPEG APP_9 Format For NITF JPEG Frame Level Overhead Data (TBR)

Offset	Field Value	Field Name	Length	Comments
0	0xFFE9	APP_9	2	NITF/JPEG
2	(See note 1)	L_P	2	Segment Length (2+length of application data)
4	0x02	APP_9 Type	1	Describes whether this app is for the image block, scan or frame portion of JPEG data (2 is frame)
5	0x01	ECC Type Number	1	0-No ECC; 1-RS-ECC; 2-BCH ECC
6	(See note 3)	Offset for SOI	4	Number of Bytes from SOI
10	(See note 3)	Size of Previous Image Block	4	Size in Bytes of Previous Image Block (See note 4)
14	(See note 3)	Size of Current Image Block	4	Size in Bytes of Current Image Block (See note 4)
18	0x464543435353594E	Start ECC SYNC Code	8	Start of ECC code for Frame ECC (FECCSSYN)
26	N/A	ECC Codes	(See note 2)	contains binary ecc codes

Notes:

1.) This value is a computed quantity based upon the number of ECC codes resulting from application of the ECC algorithm to the JPEG frame block along with the additional APP_9 application segment bytes.

2.) This value is a computed quantity based upon the number of ECC codes resulting from application of the ECC algorithm to the JPEG frame block.

3.) Unsigned binary integer format of appropriate length.

4.) Size in bytes is defined by the difference in bytes between the EOI marker minus the SOI marker position plus one of the given image block position.

Table XXIII NITF-JPEG APP_9 Format For NITF JPEG Scan Level Overhead Data (TBR)

Offset	Field Value	Field Name	Length	Comments
0	0xFFE9	APP_9	2	NITF/JPEG
2	(See note 1)	L_P	2	Segment Length (2+length of application data)
4	0x03	APP_9 Type	1	Describes whether this app is for the image block, scan or frame portion of JPEG data (3 is scan)
5	0x01	ECC Type Number	1	0-No ECC; 1-RS-ECC; 2-BCH ECC
6	(See note 3)	Offset for SOI	4	Number of Bytes from SOI
10	(See note 3)	Size of Previous Image Block	4	Size in Bytes of Previous Image Block (See note 4)
14	(See note 3)	Size of Current Image Block	4	Size in Bytes of Current Image Block (See note 4)
18	0x534543435353594E	Start ECC SYNC Code	8	Start of ECC code for Frame ECC (SECCSSYN)
26	N/A	ECC Codes	(See note 2)	contains binary ecc codes

Notes:

1.) This value is a computed quantity based upon the number of ECC codes resulting from application of the ECC algorithm to the JPEG scan block along with the additional APP_9 application segment bytes.

2.) This value is a computed quantity based upon the number of ECC codes resulting from application of the ECC algorithm to the JPEG scan block.

3.) Unsigned binary integer format of appropriate length.

4.) Size in bytes is defined by the difference in bytes between the EOI marker minus the SOI marker position plus one of the given image block position.

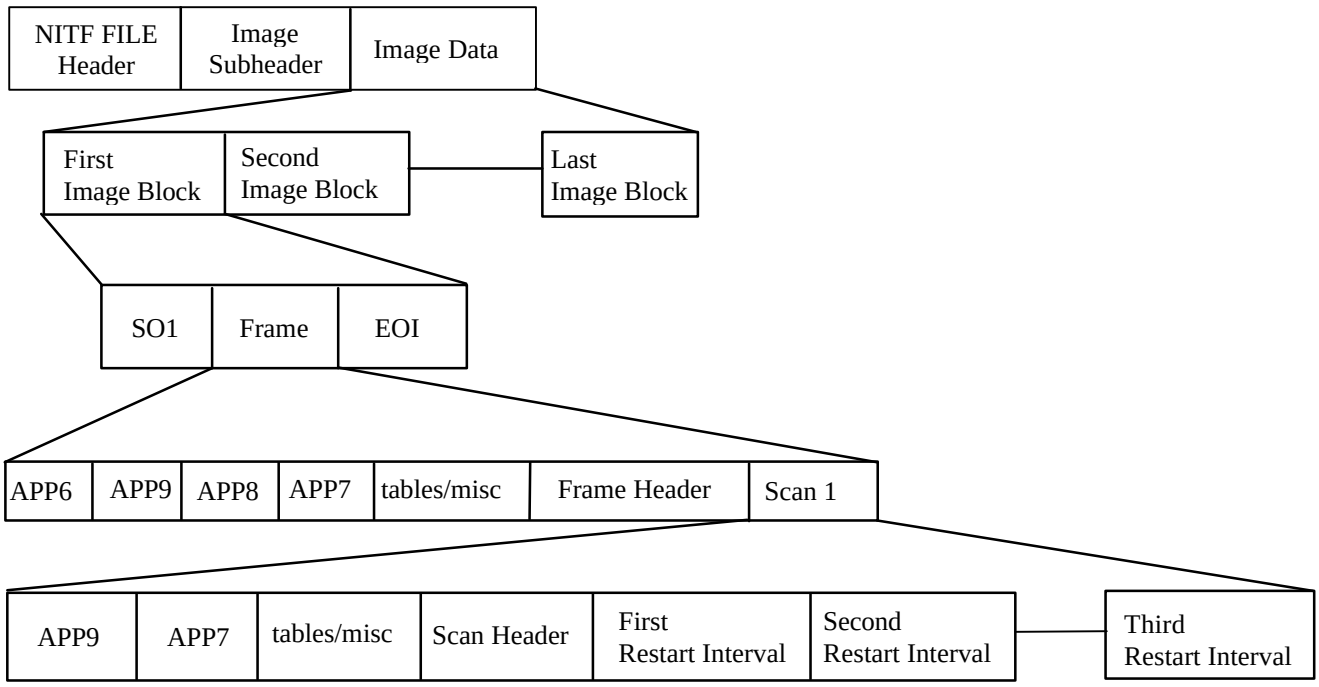

FIGURE XXI. NITF 12 BIT JPEG/DCT MULTIPLE BLOCK FILE STRUCTURE WITH FEC (IMODE=S)

The FEC code utilized to error correct the JPEG header and misc tables is based upon MIL-STD-2045-44500(TACO2), paragraph 5.4.2.1, and is termed the FEC-1 code.

For purposes of applying the FEC-1 code, the JPEG header and misc tables segments shall be logically separated into virtual datagrams of 152 bytes in length.

If the JPEG header and misc. tables segments do not break evenly into whole virtual datagrams, the remaining data, i.e. the last virtual datagram shall have fewer that 152 message bytes, but will still result in a 10 byte FEC.

The code is applied to the 152 message byte virtual datagrams comprising the overhead data.

The resultant Reed-Solomon check bytes (10 bytes per 152 message bytes) are stored separately in the APP_9 application data segment. *The storing of the check bytes separately from the datagrams is different that what paragraph 5.4.2.1 of the FEC-1 code in MIL-STD-2045-44500 (TACO2) describes.*

The resulting length of the APP_9 application segment will vary dependent upon the length of the JPEG header and misc. Tables (excluding the APP_9 application segment) as the number of FEC codes directly affect the length of the APP_9 field.

Tables XXII and XXIII contain the formats for the APP_9 segment.